WHAT IS CLAIMED IS:

1.   A method for implementing a locking mechanism to control access to a shared resource, comprising:

    receiving a request to access the shared resource;

    determining whether a first file has a first name;

    renaming the first file to a second name if the first file has the first name;

    updating a second file to indicate the received request in a queue of requests to the shared resource if the first file is renamed to the second name, wherein an ordering of the requests in the queue is used to determine whether access to the shared resource is granted to the request; and

    renaming the first file to the first name after the second file is updated.

2.   The method of claim 1, wherein updating the second file further comprises:

    reading and buffering a checksum from the second file before updating the second file;

    copying the second file to a temporary file;

    updating the temporary file to indicate the received request; and

    if the buffered checksum and checksum in the second file match, then replacing the second file with the updated temporary file.

3.   The method of claim 2, further comprising:

    calculating a revised checksum from the updated temporary file; and

    including the revised checksum in the updated temporary file before replacing the second file with the updated temporary file.

1   4.   The method of claim 2, further comprising:

2   returning retry if the buffered checksum and checksum in the second file do not

3 match, wherein the second file is not replaced with the updated temporary file if there is

4 no match.

1   5.   The method of claim 1, wherein updating the second file further

2 comprises:

3   reading and buffering a checksum calculated from the second file before updating

4 the second file, wherein the checksum is maintained in both the first file and the second

5 file;

6   copying the second file to a temporary file, wherein the temporary file is updated

7 to indicate the received request; and

8   if the buffered checksum and the checksum in both the first file and second file all

9 match, then replacing the second file with the updated temporary file.

1   6.   The method of claim 5, further comprising:

2   calculating a revised checksum from the updated temporary file;

3   including the revised checksum in the updated temporary file before replacing the

4 second file with the temporary file; and

5   including the revised checksum in the first file before renaming the first file to the

6 first name.

1   7.   The method of claim 1, wherein a lease data structure indicates at least

2 one request indicated in the queue in the second file granted access to the shared resource

3 and, for each request granted access to the shared resource, a lease time during which the

4 request is granted access to the shared resource, further comprising:

5   determining whether one request in the queue is permitted access to the shared

6 resource based on the ordering of the requests in the queue;

7   updating the lease data structure to indicate the determined request and the lease

8   time if the determined request is permitted access to the shared resource; and

9   returning a message to the determined request indicating that access to the shared

10   resource is granted and the lease time during which access is granted.

1   8.   The method of claim 7, wherein the request to access is for exclusive

2   access to the shared resource, wherein determining whether one request in the queue is

3   permitted access to the shared resource further comprises:

4   determining the request following the request at a top of the queue after

5   determining that the lease time has expired;

6   removing the request at the top of the queue;

7   updating the queue to indicate the determined request at the top of the queue; and

8   updating the lease data structure to identify the determined request and set a new

9   lease time for the determined request in the lease data structure during which the request

10   has exclusive access to the shared resource.

1   9.   The method of claim 8, wherein the queue and lease structure are updated

2   after renaming the first file from the first name to the second name.

1   10.   The method of claim 7, wherein the lease data structure comprises a file

2   and wherein updating the lease data structure further comprises:

3   reading and buffering a checksum from the lease data structure;

4   copying the lease data structure to a temporary lease file, wherein the temporary

5   lease file is updated to indicate the determined request and the new lease time; and

6   if the buffered checksum and checksum in the lease data structure match, then

7   replacing the lease data structure with the temporary lease file.

1   11.   The method of claim 7, wherein the request to access is for non-exclusive

2   access to the shared resource and wherein the lease data structure indicates a number of

3   requests allowed simultaneous access to the shared resource, wherein determining

4   whether one considered request in the queue is permitted access to the shared resource

5   further comprises:

6   determining whether a number of current readers is less than the allowed readers;

7   determining whether less than the number of current readers precedes the

8   considered request in the queue;

9   updating the lease data structure to identify the considered request and set a new

10   lease time for the considered request to have non-exclusive access to the shared resource

11   if the number of current readers is less than the allowed readers and less than the number

12   of current readers precedes the considered request in the queue; and

13   incrementing the number of current readers after updating the lease data structure

14   to indicate the considered request.

1   12.   The method of claim11, further comprising:

2   removing one request for non-exclusive access from the queue whose lease time

3   has expired;

4   removing indication from the lease data structure of the request removed from the

5   queue; and

6   decrementing a field in the lease data structure indicating the number of current

7   readers.

1   13.   The method of claim 11, wherein the queue is capable of including entries

2   for exclusive and non-exclusive access to the shared resource, and wherein the lease data

3   structure is updated to indicate the considered request and set a new least time for the

4   considered request to have non-exclusive access to the shared resource if there is no

5   exclusive access request between a top of the queue and the considered request.

1   14.   The method of claim 7, further comprising:

2   returning a retry message to the request indicating to retry the request to access

3 the shared resource and the lease time if the request is determined not to be permitted

4 access to the shared resource, wherein the request is retried after the lease time has

5 expired.

1   15.   The method of claim 1, further comprising:

2   receiving a request to remove an entry from the queue;

3   determining whether the first file has the first name;

4   renaming the first file to the second name if the first file has the first name;

5   updating the second file to indicate that the request is removed from the queue;

6 and

7   renaming the first file to the first name after the second file is updated.

1   16.   The method of claim 1, wherein the locking mechanism is capable of

2 being executed on multiple operating system platforms, and wherein the steps of

3 renaming and updating correspond to native operating system commands implemented

4 across operating system platforms.

1   17.   The method of claim 16, wherein the locking mechanism is implemented

2 in a cross-platform computer programming language that is called by applications

3 seeking to access the shared resource.

1   18.   The method of claim 17, wherein the cross-platform computer

2 programming language comprises Java.

1   19.   The method of claim 1, wherein the shared resource comprises a data

2 structure and wherein the access comprises one of read or write access to the shared data

3 structure.

1   20.   A system for implementing a locking mechanism to control access to a

2   shared resource, comprising:

3   a file system;

4   a processor in communication with the first computer readable medium and file

5   system;

6   code implemented in the computer readable medium executed by the processor to

7   cause the processor to perform:

8   (i) receiving a request to access the shared resource;

9   (ii) determining whether a first file in the file system has a first name;

10   (iii) renaming the first file to a second name if the first file has the first

11   name;

12   (iv) updating a second file in the file system to indicate the received

13   request in a queue of requests to the shared resource if the first file is renamed to

14   the second name, wherein an ordering of the requests in the queue is used to

15   determine whether access to the shared resource is granted to the request; and

16   (v) renaming the first file to the first name after the second file is updated.

1   21.   The system of claim 20, further comprising:

2   a computer readable medium,

3   wherein the code for updating the second file further causes the processor to

4   perform:

5   (i) reading and buffering a checksum from the second file in the computer

6   readable medium before updating the second file;

7   (ii) copying the second file to a temporary file;

8   (iii) updating the temporary file to indicate the received request; and

9   (iv) if the buffered checksum and checksum in the second file match, then

10   replacing the second file with the updated temporary file.

1   22.   The system of claim 20, further comprising:

2   a computer readable medium;

3   a lease data structure in the computer readable medium indicating at least one

4   request indicated in the queue in the second file granted access to the shared resource

5   and, for each request granted access to the shared resource, a lease time during which the

6   request is granted access to the shared resource;

7   wherein the code further causes the processor to perform:

8   (i) determining whether one request in the queue is permitted access to the

9   shared resource based on the ordering of the requests in the queue;

10   (ii) updating the lease data structure to indicate the determined request and

11   the lease time if the determined request is permitted access to the shared resource;

12   and

13   (iii) returning a message to the determined request indicating that access

14   to the shared resource is granted and the lease time during which access is

15   granted.

1   23.   The system of claim 22, wherein the request to access is for exclusive

2   access to the shared resource, wherein the code causing the processor to determine

3   whether one request in the queue is permitted access to the shared resource further causes

4   the processor to perform:

5   determining the request following the request at a top of the queue after

6   determining that the lease time has expired;

7   removing the request at the top of the queue;

8   updating the queue to indicate the determined request at the top of the queue; and

9   updating the lease data structure to identify the determined request and set a new

10   lease time for the determined request in the lease data structure during which the request

11   has exclusive access to the shared resource.

1   24.   The system of claim 22, wherein the request to access is for non-exclusive

2 access to the shared resource and wherein the lease data structure indicates a number of

3 requests allowed simultaneous access to the shared resource, wherein the code causing

4 the processor to determine whether one considered request in the queue is permitted

5 access to the shared resource further causes the processor to perform:

6   determining whether a number of current readers is less than the allowed readers;

7   determining whether less than the number of current readers precedes the

8 considered request in the queue;

9   updating the lease data structure to identify the considered request and set a new

10 lease time for the considered request to have non-exclusive access to the shared resource

11 if the number of current readers is less than the allowed readers and less than the number

12 of current readers precedes the considered request in the queue; and

13   incrementing the number of current readers after updating the lease data structure

14 to indicate the considered request.

1   25.   The system of claim 20, wherein the code further causes the processor to

2 perform:

3   receiving a request to remove an entry from the queue;

4   determining whether the first file has the first name;

5   renaming the first file to the second name if the first file has the first name;

6   updating the second file to indicate that the request is removed from the queue;

7 and

8   renaming the first file to the first name after the second file is updated.

1   26.   The system of claim 20, wherein the locking mechanism is capable of

2 being executed on multiple operating system platforms, and wherein the steps of

3 renaming and updating correspond to native operating system commands implemented

4 across operating system platforms.

1   27.   The system of claim 20, wherein the shared resource comprises a data

2 structure and wherein the access comprises one of read or write access to the shared data

3 structure.

1   28.   A system for implementing a locking mechanism to control access to a

2 shared resource, comprising:

3   a file system implemented including a first file and a second file;

4   means for receiving a request to access the shared resource;

5   means for determining whether the first file has a first name;

6   means for renaming the first file to a second name if the first file has the first

7 name;

8   means for updating the requests to the shared resource if the first file is renamed

9 to the second name, wherein an ordering of the requests in the queue is used to determine

10 whether access to the shared resource is granted to the request; and

11   means renaming the first file to the first name after the second file is updated.

1   29.   The system of claim 28, further comprising:

2   a computer readable medium;

3   wherein the means for updating the second file further performs:

4   (i) reading and buffering a checksum from the second file in the computer

5   readable medium before updating the second file;

6   (ii) copying the second file to a temporary file in the file system;

7   (iii) updating the temporary file to indicate the received request; and

8   (iv) if the buffered checksum and checksum in the second file match, then

9   replacing the second file with the updated temporary file.

1   30.   The system of claim 28, further comprising:

2   a computer readable medium;

3   wherein the means for updating the second file further performs:

4   (i) reading and buffering a checksum in the computer readable medium

5   calculated from the second file before updating the second file, wherein the

6   checksum is maintained in both the first file and the second file;

7   (ii) copying the second file to a temporary file in the file system, wherein

8   the temporary file is updated to indicate the received request; and

9   (iv) if the buffered checksum and the checksum in both the first file and

10   second file all match, then replacing the second file with the updated temporary

11   file.

1   31.   The system of claim 28, further comprising:

2   a computer readable medium;

3   a lease data structure in the compute readable medium that indicates at least one

4 request indicated in the queue in the second file granted access to the shared resource

5 and, for each request granted access to the shared resource, a lease time during which the

6 request is granted access to the shared resource;

7   means for determining whether one request in the queue is permitted access to the

8 shared resource based on the ordering of the requests in the queue;

9   means for updating the lease data structure to indicate the determined request and

10 the lease time if the determined request is permitted access to the shared resource; and

11   means for returning a message to the determined request indicating that access to

12 the shared resource is granted and the lease time during which access is granted.

1   32.   The system of claim 31, wherein the request to access is for exclusive

2 access to the shared resource, wherein the means for determining whether one request in

3 the queue is permitted access to the shared resource further performs:

4   determining the request following the request at a top of the queue after

5 determining that the lease time has expired;

6   removing the request at the top of the queue;

7   updating the queue to indicate the determined request at the top of the queue; and

8   updating the lease data structure to identify the determined request and set a new

9   lease time for the determined request in the lease data structure during which the request

10   has exclusive access to the shared resource.

1   33.   The system of claim 31, wherein the lease data structure comprises a file

2   and wherein the means for updating the lease data structure further performs:

3   reading and buffering a checksum from the lease data structure;

4   copying the lease data structure to a temporary lease file, wherein the temporary

5   lease file is updated to indicate the determined request and the new lease time; and

6   if the buffered checksum and checksum in the lease data structure match, then

7   replacing the lease data structure with the temporary lease file.

1   34.   The system of claim 31, wherein the request to access is for non-exclusive

2   access to the shared resource and wherein the lease data structure indicates a number of

3   requests allowed simultaneous access to the shared resource, wherein the means for

4   determining whether one considered request in the queue is permitted access to the

5   shared resource further performs:

6   determining whether a number of current readers is less than the allowed readers;

7   determining whether less than the number of current readers precedes the

8   considered request in the queue;

9   updating the lease data structure to identify the considered request and set a new

10   lease time for the considered request to have non-exclusive access to the shared resource

11   if the number of current readers is less than the allowed readers and less than the number

12   of current readers precedes the considered request in the queue; and

13   incrementing the number of current readers after updating the lease data structure

14   to indicate the considered request.

1   35.   The system of claim 31, further comprising:

2   means for returning a retry message to the request indicating to retry the request

3 to access the shared resource and the lease time if the request is determined not to be

4 permitted access to the shared resource, wherein the request is retried after the lease time

5 has expired.

1   36.   The system of claim 28, further comprising:

2   means for receiving a request to remove an entry from the queue;

3   means for determining whether the first file has the first name;

4   means for renaming the first file to the second name if the first file has the first

5 name;

6   means for updating the second file to indicate that the request is removed from the

7 queue; and

8   means for renaming the first file to the first name after the second file is updated.

1   37.   The system of claim 28, wherein the locking mechanism is capable of

2 being executed on multiple operating system platforms, and wherein the means for

3 renaming and updating utilize native operating system commands implemented across

4 operating system platforms.

1   38.   The system of claim 37, wherein the locking mechanism is implemented

2 in a cross-platform computer programming language that is called by applications

3 seeking to access the shared resource.

1   39.   The system of claim 28, wherein the shared resource comprises a data

2 structure and wherein the access comprises one of read or write access to the shared data

3 structure.

1   40.   An article of manufacture including code for implementing a locking

2 mechanism to control access to a shared resource, wherein the code causes operations to

3 be performed comprising:

4   receiving a request to access the shared resource;

5   determining whether a first file has a first name;

6   renaming the first file to a second name if the first file has the first name;

7   updating a second file to indicate the received request in a queue of requests to

8 the shared resource if the first file is renamed to the second name, wherein an ordering of

9 the requests in the queue is used to determine whether access to the shared resource is

10 granted to the request; and

11   renaming the first file to the first name after the second file is updated.

1   41.   The article of manufacture of claim 40, wherein updating the second file

2 further comprises:

3   reading and buffering a checksum from the second file before updating the second

4 file;

5   copying the second file to a temporary file;

6   updating the temporary file to indicate the received request; and

7   if the buffered checksum and checksum in the second file match, then replacing

8 the second file with the updated temporary file.

1   42.   The article of manufacture of claim 41, further comprising:

2   calculating a revised checksum from the updated temporary file; and

3   including the revised checksum in the updated temporary file before replacing the

4 second file with the updated temporary file.

1   43.   The article of manufacture of claim 41, further comprising:

2   returning retry if the buffered checksum and checksum in the second file do not

3   match, wherein the second file is not replaced with the updated temporary file if there is

4   no match.

1   44.   The article of manufacture of claim 40, wherein updating the second file

2   further comprises:

3   reading and buffering a checksum calculated from the second file before updating

4   the second file, wherein the checksum is maintained in both the first file and the second

5   file;

6   copying the second file to a temporary file, wherein the temporary file is updated

7   to indicate the received request; and

8   if the buffered checksum and the checksum in both the first file and second file all

9   match, then replacing the second file with the updated temporary file.

1   45.   The article of manufacture of claim 44, further comprising:

2   calculating a revised checksum from the updated temporary file;

3   including the revised checksum in the updated temporary file before replacing the

4   second file with the temporary file; and

5   including the revised checksum in the first file before renaming the first file to the

6   first name.

1   46.   The article of manufacture of claim 40, wherein a lease data structure

2   indicates at least one request indicated in the queue in the second file granted access to

3   the shared resource and, for each request granted access to the shared resource, a lease

4   time during which the request is granted access to the shared resource, further

5   comprising:

6   determining whether one request in the queue is permitted access to the shared

7   resource based on the ordering of the requests in the queue;

8   updating the lease data structure to indicate the determined request and the lease

9   time if the determined request is permitted access to the shared resource; and

10   returning a message to the determined request indicating that access to the shared

11   resource is granted and the lease time during which access is granted.

1   47.   The article of manufacture of claim 46, wherein the request to access is for

2   exclusive access to the shared resource, wherein determining whether one request in the

3   queue is permitted access to the shared resource further comprises:

4   determining the request following the request at a top of the queue after

5   determining that the lease time has expired;

6   removing the request at the top of the queue;

7   updating the queue to indicate the determined request at the top of the queue; and

8   updating the lease data structure to identify the determined request and set a new

9   lease time for the determined request in the lease data structure during which the request

10   has exclusive access to the shared resource.

1   48.   The article of manufacture of claim 47, wherein the queue and lease

2   structure are updated after renaming the first file from the first name to the second name.

1   49.   The article of manufacture of claim 46, wherein the lease data structure

2   comprises a file and wherein updating the lease data structure further comprises:

3   reading and buffering a checksum from the lease data structure;

4   copying the lease data structure to a temporary lease file, wherein the temporary

5   lease file is updated to indicate the determined request and the new lease time; and

6   if the buffered checksum and checksum in the lease data structure match, then

7   replacing the lease data structure with the temporary lease file.

1   50.   The article of manufacture of claim 56, wherein the request to access is for

2   non-exclusive access to the shared resource and wherein the lease data structure indicates

3   a number of requests allowed simultaneous access to the shared resource, wherein

4   determining whether one considered request in the queue is permitted access to the

5   shared resource further comprises:

6   determining whether a number of current readers is less than the allowed readers;

7   determining whether less than the number of current readers precedes the

8   considered request in the queue;

9   updating the lease data structure to identify the considered request and set a new

10   lease time for the considered request to have non-exclusive access to the shared resource

11   if the number of current readers is less than the allowed readers and less than the number

12   of current readers precedes the considered request in the queue; and

13   incrementing the number of current readers after updating the lease data structure

14   to indicate the considered request.

1   51.   The article of manufacture of claim 50, further comprising:

2   removing one request for non-exclusive access from the queue whose lease time

3   has expired;

4   removing indication from the lease data structure of the request removed from the

5   queue; and

6   decrementing a field in the lease data structure indicating the number of current

7   readers.

1   52.   The article of manufacture of claim 50, wherein the queue is capable of

2   including entries for exclusive and non-exclusive access to the shared resource, and

3   wherein the lease data structure is updated to indicate the considered request and set a

4   new least time for the considered request to have non-exclusive access to the shared

5   resource if there is no exclusive access request between a top of the queue and the

6   considered request.

1   53.   The article of manufacture of claim 46, further comprising:

2   returning a retry message to the request indicating to retry the request to access

3   the shared resource and the lease time if the request is determined not to be permitted

4   access to the shared resource, wherein the request is retried after the lease time has

5   expired.

1   54.   The article of manufacture of claim 40, further comprising:

2   receiving a request to remove an entry from the queue;

3   determining whether the first file has the first name;

4   renaming the first file to the second name if the first file has the first name;

5   updating the second file to indicate that the request is removed from the queue;

6   and

7   renaming the first file to the first name after the second file is updated.

1   55.   The article of manufacture of claim 40, wherein the locking mechanism is

2   capable of being executed on multiple operating system platforms, and wherein the steps

3   of renaming and updating correspond to native operating system commands implemented

4   across operating system platforms.

1   56.   The article of manufacture of claim 55, wherein the locking mechanism is

2   implemented in a cross-platform computer programming language that is called by

3   applications seeking to access the shared resource.

1   57.   The article of manufacture of claim 56, wherein the cross-platform

2   computer programming language comprises Java.

1   58.   The article of manufacture of claim 40, wherein the shared resource

2   comprises a data structure and wherein the access comprises one of read or write access

3   to the shared data structure.

1   59.   A computer readable medium including data structures for implementing a

2   locking mechanism to control access to a shared resource, comprising:

3   a first file having one of a first name or a second name, wherein in response to

4   receiving a request to access the shared resource, the first file is renamed to the second

5   name if the first file has the first name; and

6   a second file indicating a queue of requests to the shared resource, wherein an

7   ordering of the requests in the queue is used to determine whether access to the shared

8   resource is granted to the request, and wherein the second file is updated to indicate the

9   received request in the queue when the first file is renamed to the second name, wherein

10   the first file is renamed to the first name after the second file is updated.

1   60.   The computer readable medium of claim 59, further comprising:

2   a checksum field in the second file, wherein the checksum is buffered before

3   updating the second file; and

4   a temporary file, wherein the second file is copied to the temporary file and the

5   temporary file is updated to indicate the received request, wherein the second file is

6   replaced with the updated temporary file if the buffered checksum and checksum in the

7   second file match.

1   61.   The computer readable medium of claim 59, further comprising:

2   a lease data structure indicating at least one request indicated in the queue in the

3   second file granted access to the shared resource and, for each request granted access to

4   the shared resource, a lease time during which the request is granted access to the shared

5   resource, wherein determining whether one request in the queue is permitted access to the

6   shared resource is based on the ordering of the requests in the queue, and wherein the

7   lease data structure is updated to indicate the determined request and the lease time if the

8   determined request is permitted access to the shared resource.

1   62.   The computer readable medium of claim 61, wherein the lease data

2   structure comprises a file, further comprising:

3   a checksum field within the lease data structure, wherein the checksum from the

4   checksum field is buffered when updating the lease data structure; and

5   a temporary lease file, wherein updating the lease data structure involves copying

6   the lease data structure to the temporary lease file, wherein the temporary lease file is

7   updated to indicate the determined request following the request at the tope of the queue

8   and the new lease time, wherein if the buffered checksum and checksum in the lease data

9   structure match, then the lease data structure is replaced with the temporary lease file.

1   63.   The computer readable medium of claim 61, wherein the request to access

2   is for non-exclusive access to the shared resource, further comprising:

3   a number of allowed readers allowed simultaneous access to the shared resource

4   indicated in the lease data structure, wherein the lease data structure is updated to identify

5   the considered request and set a new lease time for the considered request to have non-

6   exclusive access to the shared resource if a number of current readers is less than the

7   allowed readers and less than the number of current readers preceding the considered

8   request in the queue, wherein the number of current readers is incremented after updating

9   the lease data structure to indicate the considered request.